Big data privacy preservation using K-Anonymization and l-Diversity

Priyanka Gawali[1], Dhananjay Gawali[2]
[1, 2] Department of CSE
[1] Dr. DYPSOET, Lohegaon Pune, India
[2] New Art Commerce & Science College, Shevgaon,India

Abstract- *Classification is a fundamental problem in data analysis. Training a classifier requires to get a large collection of data. Releasing person-specific data in its most specific state poses a threat to individual privacy. This paper presents a practical and productive algorithm for determining a abstract version of data that masks sensitive information and remains useful for standardizing structuring. The analysis of data is implemented by specializing or detailing the level of information in a top-down and bottom-up manner until a minimum privacy requirement is compromised. This top-down and bottom-up specialization is practical and efficient for handling both definitive and continuous attributes. Our method exploits the scenario that data usually contains redundant structures for classification. While generalization may remove few structures, other structures originate to help. Our results show that standard of classification can be preserved even for highly prohibitive privacy requirements. This work has big applications to both public and private sectors that share information for mutual advantage and productivity. Experiments on real-life data show that the quality of classification can be preserved even for highly restrictive anonymity requirements.*

Keywords**-** Data anonymization; l-diversity; privacy preservation; hadoop.

I. INTRODUCTION

With the development of Internet technology and data processing technology, a large number of data associated with individuals, such as demographic data, patient medical data is widely collected and published by government departments and research institutions. However, these data may contain private information of individuals, and the presence of a large number of data causes the widespread use of data mining tools, so the protection of personal privacy information is of great concern [1]. So, for an anonymization of the data, a variety of anonymous principles and anonymous technology is proposed, including generalization [2,3] used in k-anonymity[3] model and Anatomy[4] the decomposition technique based on lossy connections used in l-diversity[5].

II. RELATED WORK

In recent years, the anonymous protection of sensitive attributes in data publication has been concerned by many researchers. The study of literature [6,7] has shown that the optimal data anonymous (i.e. achieving anonymity on sensitive attributes, while making information loss minimization) is a NP-hard problem. Focusing on how to reduce the loss of information in anonymous protection and improve the practicality of published data, many data anonymous technology have been put forward, such as generalization and decomposition based on lossy connections.

Local re-encoding is a encoded form of generalization which retains more information. Literature [5,8] indicates the inadequacies of generalize technique in privacy protection, especially, generalization loses a lot of information when dealing with high dimensional data.

Compared with generalization, decomposition based on lossy connections also has several limitations. It is necessary to separate sensitive data attribute and quasi-identifier attribute clearly. But in real life data, it is impossible to distinguish sensitive and quasi-identifier attribute obviously. Study [3] showed that about 87 percent of U.S. residents can be uniquely identified through a quasi-identifier.

Literature [9] proposed a new decomposition technique based on lossy connections-slicing. Compared with the previous decomposition techniques, this method ensures the security of the data better. But the downside of the method is a hard clustering process of data attributes, which reduces data availability, what's more, the time complexity of packet processing algorithm is too high.

III. OBJECTIVES

The main objectives of this paper introduce: Scalability and Parallel computation.

Scalability: Provides high scalability by using job level and task level parallelization. Job level parallelization means that multiple MapReduce jobs can be executed simultaneously to make full use of cloud infrastructure assets. Combined with cloud, MapReduce becomes more powerful and elastic can offer infrastructure resources on demand. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data splits.

Parallel computation: To make full use of the parallel capability of MapReduce , specializations needed in an anonymization process are divided into phases.

IV. PROPOSED SYSTEM

This method we analyze the scalability problem of existing TDS approaches while handling large-scale data sets on HADOOP platform.

1. Top-Down Specialization

TDS is repeated process which is starting from the topmost domain values in the arrangement trees of attributes. Each round of iteration consists of 3 main steps. Finding the best specialization, performing specialization and updating values of the search metric for the next round [10]. Such a process of TDS is repeated until k-anonymity is violated, to description for the maximum data is going to utilize in that. The righteousness of a specialization is measured by a search metric. In that we accept the information gain per privacy loss (IGPL), a tradeoff metric that take in mind both the privacy and information requirements, as the search metric in our approach [11]. A specialization with the highest IGPL value is regarded as best one and selected of each round.

I. Algorithm:

```
Algorithm TDS
Initialize every value in T to the top most value.
Initialize $Cut_i$ to include the top most value.
while some x $\in \cup Cut_i$ is valid and beneficial do
Find the Best specialization from $\cup Cut_i$.
Perform Best on T and update $\cup Cut_i$.
Update Score(x) and validity for x $\in \cup Cut_i$.
end while
return Ge
```

II. Direct Anonymization Algorithm DA (D,I,k,m)

```
Scan D and create count-tree
Initialize Count
For each node v in preorder count-tree traversal do
If the item of v has been generalized in Count then
Backtrack
If v is a leaf node and v.count<k then
J:=itemset corresponding to v
Find generalization of items in J that make J k-anonymous
Merge generalization rules with Count
Backtrack to longest prefix of path J,wherein no item has  been generalized in Count
Return Count
for i :=1 to Count do
Initialize count=0
Scan each transactions in Count
Separate each item in a transaction and store it in p
Increment count
for j:=1 to count do
For all g belongs Count do.
Compare each item of p with that of Count
If all items of i equal to count
Increment the r
If k equal to r then backtrack to i
else if r greater than ka then get the index position of the similar transactions
Make them NULL until ka equal to r normalized T and $\cup Cut_i$.
Else update the transactions in database.
```

II. Advantages:

Top-down approach

- Handling multiple VIDs
    - Treating all VIDs as a single VID leads to over generalization.
- Handling both categorical and continuous attributes.
    - Dynamically generate taxonomy tree for continuous attributes.
- Anytime solution
    - User may step through each specialization to determine a desired trade-off between privacy and accuracy.
    - User may stop any time and obtain a generalized table satisfying the anonymity requirement. Bottom-up approach does not support this feature.
- Scalable computation
- Hierarchically organized (top-down) architecture
- All the necessary knowledge is pre-programmed, i.e. already present - in the knowledge base.
- Analysis/computation includes creating, manipulating and linking symbols (hence propositional and predicate- calculus approach).
- "Serial executive" can be looked as the natural rule-interpreter which acts on the parallel-processing unconscious intuitive processor.
- Thus the program behaves better at relatively high-level tasks such as language processing aka NLP - it

is consistent with currently accepted theories of language acquisition which assume some high-level modularity.

2. Bottom Up Generalization

Bottom-up generalization is an iterative method from data processing to generalize the information. It is difficult to link to alternative sources even though the generalized data remains helpful for classification. The generalization house is mere by a data structure of generalizations. A key is at each iteration the best generalization is distinguished to climb up the hierarchy. The bottom-up generalization converts the specific data to less specific but semantically consistent data for privacy preservation and also they focused on two main problems, scalability and quality. The scalability problem was addressed by a unique data structure to focus on pretty good generalizations. The same quality is achieved by the proposed system however far better measurability compared to existing solutions. Our current algorithm has the likelihood of obtaining stuck at a neighborhood optimum by greedily hill climbs to a k-anonymity state.

Algorithm presents the general idea of bottom-up generalization method. It begins the generalization from the raw data table T. At each iteration, the algorithm greedily selects the Best generalization g that minimizes the information loss and maximizes the privacy gain. This intuition is captured by the information metric ILPG(g) = IL(g)/PG(g). Then, the algorithm performs the generalization child(Best) → Best on the table T , and repeats the iteration until the table T satisfies the given k-anonymity requirement.

Algorithm 3.1.1 Bottom-Up Generalization

```
1. while T does not satisfy a given k-anonymity requirement do
2. for all generalization g do
3. compute ILPG(g);
4. end for
5. find the Best generalization;
6. generalize T by Best;
7. end while
8. output T;
```

Let A(QID) and Ag(QID) be the minimum anonymity counts in T before and after the generalization g. Given a data table T, there are many possible generalizations that can be performed. Yet, most generalizations g in fact does not affect the minimum anonymity count. In other words, A (QID) = Ag(QID). Thus, to facilitate efficiently choosing a generalization g, there is no need to consider all generalizations. Indeed, we can focus only on the “critical generalizations.” DEFINITION 3.1: A generalization g is critical if Ag (QID) > A(QID). Wang et al. [1] made several observations

Algorithm 3.1.2 Bottom-Up Generalization

```
1. while T does not satisfy a given k-anonymity requirement do
2. for all critical generalization g do
3. compute Ag (QID);
4. end for
5. find the Best generalization;
6. generalize T by Best;
7. end while
8. output T;
```

To improve the efficiency of the generalization operation, propose a data structure, called Taxonomy Encoded Anonymity (TEA) index for QID = D1. . . Dm. TEA is a tree of m levels. The ith level represents the current value for Dj. Each root-to-leaf path represents a qid value in the current data table, with a(qid) stored at the leaf node. the TEA index links up the qids according to the generalizations that generalize them. When a generalization g is applied, the TEA index is updated by adjusting the qids linked to the generalization of g. The purpose of this index is to prune the number of candidate generalizations to no more than |QID| at each iteration, where |QID| is the number of attributes in QID. For a generalization g: child (v) → v, a segment of g is a maximal set of sibling nodes, {s1. . . st}, such that {s1, . . . , st} & child(v), where t is the size of the segment. All segments of g are linked up. A qid is generalized by a segment if the qid contains a value in the segment. A segment of g represents a set of sibling nodes in the TEA.

V. EXPERIMENTAL RESULTS

Proposed system is used to securely publish data and maintain the privacy of sensitive attribute. Now a days there are many encryption algorithms available which give maximum security to attribute. But their computation time is high as compare to this system as shown in graph. We compare proposed algorithm with blowfish encryption algorithm as its performance is better than all other encryption algorithms. If any provider wants to send our data to other user, instead of encryption algorithm he can use slicing algorithm. For small scope system like hospital management system where SA is disease or banking sector where SA will be balance of customer. On above 25 records of input, Figure 1 and Figure 2 shows computation time between slicing and encryption algorithm. This shows the performance of the system i.e CPU usage in millisecond of the system on which it runs.

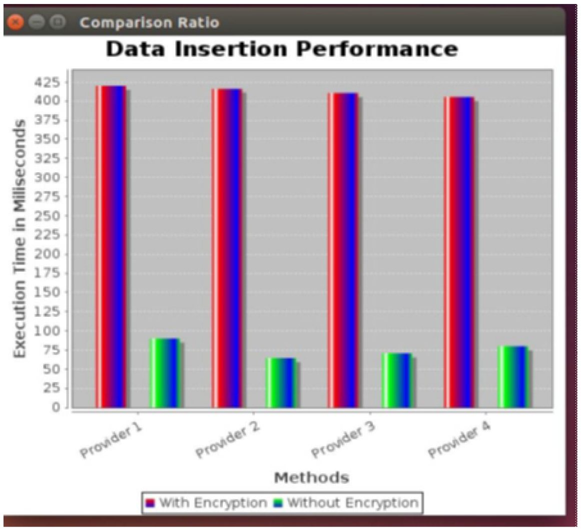

Figure 1 Data Insertion Performance with Existing and Proposed System

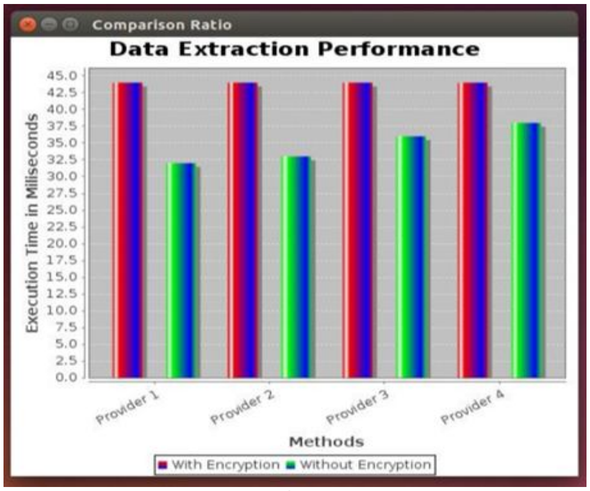

Figure 2 Data extraction Performance with Existing and Proposed System

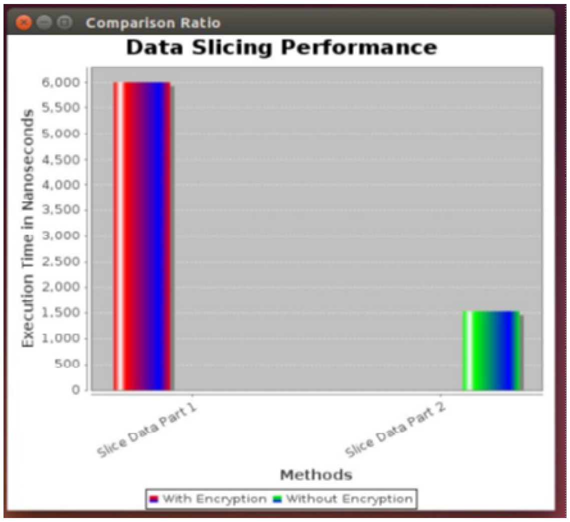

Figure 3 Data slicing performance with Existing and Proposed System

VI. CONCLUSION

Privacy preserving data analysis and data publishing are becoming serious problems in today's ongoing world. That's why different approaches of data anonymization techniques are proposed. There are various anonymization techniques present and they mainly focused on k-anonymity which comprises of both generalization and suppression. The generalization algorithms and its implementation for protecting the privacy of data used mainly for data analysis. In particular, the paper presented a bottom-up generalization for transforming specific data to less specific but semantically consistent data for privacy protection. TDS approach using MapReduce are applied on hadoop to data anonymization and deliberately designed a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way.

REFERENCES

[1] V.S.Iyengar.Transforming data to satisfy privacy constraints[C].In the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining., Alberta, Canada: ACM Press Press, 2002.279-288.

[2] P.Samarati.Protecting Respondent □ s Privacy in Microdata Release [J], IEEE Trans. Knowledge and Data Eng.Nov./Dec. 2001, 13(6):1010-1027.

[3] Sweeney L. *K*-anonymity: a model for protecting privacy [J]. International Journal of Uncertainty, Fuzziness and Knowledge based Systems.2002, 10(5):557-570.

[4] X.Xiao Y.Tao, Anatomy:Simple and Effective Privacy Preservation[C],Proc.Int□l Conf.Very Large Data Bases(VLDB), 2006: 139-150

[5] A. Machanavajjhala, J. Gehrke, and D. Kifer. L-diversity: Privacy Beyond k-anonymity.in:22nd International Conference on Data Engineering (ICDE □ 06). Atlanta, Georgia, USA: IEEE Computer Society Press. 2006, 24-36.

[6] Aggarwal G□Feder T□Kenihapadi K□et al□Achieving anonymity via clustering [C]. In: Vansummeren S,ed.Proc.of the 25th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems .New York: ACM,2006:153-162.

[7] Meyerson A,Williams R.On the Complexity of optimal k-anonymity [C]. In:Deutsch A,ed.Proc.of the 23rd

ACM SIGACT-SIGMODSIGART Symp.on Principles of Database Systems.New York: ACM, 2004 :223-228.

[8] C.Aggarwal, On k-Anonymity and the curse of Dimensionality[C], Proc.Int□l Conf.Very Large Data Bases (VLDB), 2005:901-909.

[9] Tiancheng Li,Ninghui Li,Jian Zhang. Slincing : A New Approach for Privacy Preserving Data Publishing [J].IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2012, 24(3):561-574.

[10] MicrosoftHealthVault, http://www.microsoft.com/health/ww/ roducts/Pages/ healthvault.aspx, accessed on: Jan. 05, 2013.

[11] Benjamin C. M. Fung, Ke Wang, Philip S. Yu, “Top-DownSpecialization for Information and Privacy Preservation”,Proceedings of the 21st International Conference on DataEngineering (ICDE 2005), 1084-4627/05 $20.00 © 2005 IEEE.

[12] Jing Yang , Ziyun Liu , yangyue ,Jianpei Zhang “A Data Anonymous Method based on Overlapping Slicing” Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design 978-1-4799-3776-9/14/$31.00 ©2014 IEEE